


# DAS MAGAZIN FÜR COMPUTER-FANS

**Tabula Rasa:** Zahlen am laufenden Band – Tabellenkalkulation der Sonderklasse

**Sir-Compact III:** verdichtet Basic- und Assemblerprogramme um mehr als die Hälfte!

**Superfrac 64:** Bizarre Landschaften – Mathematik mit Romantik

**Flexgrid:** Hexenküche der Intro-Programmierer

**Blox!:** Raffiniertes Denkspiel für schlaue Computerfüchse

**System-V:** Mehr Spaß beim Vokabelpauken!

## Impressum

**Chefredakteur:** Harald Beiler (bl), verantwortlich für den redaktionellen Teil
**Redaktion:** Jörn-Erik Burkert (lb)
**Redaktionsassistenz:** Marianne Schmitt (sm)

**So erreichen Sie die Redaktion:**
Tel. 089/4613-414, Fax 089/4613-433, Btx *64064#
Hotline 089/4613-640, jd.. Mi., 15-17.00 Uhr

**Manuskripteinsendungen:** Programmlistings werden gerne von der Redaktion angenommen. Sie müssen frei sein von Rechten Dritter. Sollten sie auch an anderer Stelle zur Veröffentlichung oder gewerblichen Nutzung angeboten worden sein, so muß das angegeben werden. Mit der Einsendung von Listings gibt der Verfasser die Zustimmung zum Abdruck in den von der MagnaMedia Verlag AG herausgegebenen Publikationen und zur Vervielfältigung der Programmlistings auf Datenträgern. Honorare nach Vereinbarung. Für unverlangt eingesandte Listings wird keine Haftung übernommen.

**Gestaltung**: Paul Dlugosch, RudolfScharl

**Anzeigenverkaufsleiter:** Regine Schmidt (828), verantwortlich für den Anzeigenteil

**Anzeigenverwaltung / Disposition:** Regina Beenken (372)
**Anzeigenpreise:** Es gilt die Preisliste Nr. 13 vom 01. 01. 1996

**So erreichen Sie die Anzeigenabteilung**
Tel. 089/4613-962, Telefax 089/4613-394

**Erscheinungsweise:** monatlich (zwölf Ausgaben im Jahr)

**Leitung Herstellung:** Klaus Buck (180)

**Druck:** Eder & Poehlmann, Grasbrunner Weg 6, 85630 Neukeferloh



**Vertriebsleitung:** Benno Gaab

**1997 CM Consumer Medien Verlag GmbH**

Die WEKA Firmengruppe GmbH & Co.KG hält 86% der Magna Media Aktien

**Geschäftsführer:** Dr. Rüdiger Hennigs, Werner Mützel

**Anschrift des Verlages:**
CM Consumer Medien Verlag GmbH
Hans-Pinsel-Str. 9a, 85540 Haar bei München

**Achtung: Die Diskette zu dieser Ausgabe ist schreibgeschützt. Um Daten darauf zu speichern, ist die Schreibkerbe rechts oben anzubringen!**






# Für Einsteiger und Profis

Warum umständlich, wenn's einfach geht? Im C 64 steckt jede Menge Potential, das man nur wecken muß. Wir bieten Ihnen eine Riesenauswahl an Tools und Tips, die nicht im Handbuch stehen.

## Interrupt verhindern

Einen »NMI« (Abkürzung für »Non-Maskable-Interrupt«) erzeugt man beim C 64 z.B. durch die Tastenkombination <RUN/STOP RESTORE>. Dann springt der Computer nämlich zum Unterprogramm im Kernel ab $FE47 (65095). Diese Adresse wurde in der Vektorentabelle in den Speicherstellen $0318/$0319 (792/793) als Low-High-Byte gespeichert. Laufende Programme werden damit abgebrochen und der Bildschirm gelöscht.

Mit ein paar POKE-Befehlen läßt sich die NMI-Leitung auf 0 Volt setzen und so abschalten:

```
5 A=PEEK(792): POKE 792,193
10 POKE 56580,0: POKE 56581,0
20 POKE 56589,129: POKE 58590,1
30 POKE 792,A
```

Was macht das Programm? In **Zeile 5** wird der Original-NMI-Vektor in der Variablen A zwischengespeichert und auf eine Speicherstelle gerichtet, in der man die Assembler-Anweisung RTI (Return from Interrupt) findet, das ist z.B. bei Adresse $FEC1 (65217) der Fall. Das Low-Byte $C1 (193) muß man in Vektor $0318 eintragen. Das ist sehr wichtig, da bei der geplanten Manipulation der CIA-Register (Zeile 10) unweigerlich ein NMI eintritt, der die Werte des CIA-Chips wieder normalisieren würde – das ganze Basic-Listing wäre für die Katz.

Jetzt muß man in **Zeile 10** den CIA-Interruptzähler auf »0« setzen und den Interrupt mit dem temporär gespeicherten Wert wieder einschalten (POKE 792, A). Der CIA2-Chip hält nun die NMI-Leitung auf »Low« (kein Strom).

Das Timing eines Hauptprogramms (z.B. Zugriffe auf den seriellen Bus) oder eines aktivierten Raster-IRQ bringt man damit keineswegs durcheinander.

So lassen sich z.B. Hintergrundaktionen realisieren. Beispiel: Ein Programm lädt von Floppy nach, während sich Sprites auf dem Bildschirm tummeln oder fetzige Musik aus dem SID-Chip ertönt.

## Sicher ist sicher ...

Der NEW-Befehl verbannt Basic-Programme aus dem Speicher. Pech, wenn man anschließend entsetzt feststellt, daß man das mühevoll entwickelte Programmprojekt nicht vorher auf Diskette gesichert hatte!

Kein Panik – wenn Sie zwischenzeitlich kein anderes Programm nachgeladen haben, läßt sich alles retten. Dazu müssen Sie im Direktmodus folgenden Basic-Einzeiler eingeben:

```
POKE 2050,8: SYS 42291: POKE 46,PEEK(35)-
PEEK(781)>253): POKE 45,PEEK(781)+2 AND
255: CLR
```

Damit holen Sie jedes Basic-Programm unversehrt wieder zurück. Es klappt auch, wenn Sie den Computer per Reset-Schalter in seinen Einschaltzustand versetzt haben.

## Wieviele Tage hat der Monat?

Viele C-64-Anwendungen (z.B. Dateiverwaltungen, Kassenbuchprogramme usw.) verlangen die Eingabe des Tagesdatums – meist im Format TTMMJJ (Tag, Monat, Jahr). Vor allem, wenn das Datum vom Programm ausgewertet wird, sollte die Eingabe hieb- und stichfest sein. Der »30. Februar« z.B. kann so manche Berechungen ins Nirwana schicken. Ein Computer kann zwar nicht beurteilen, ob das Datum stimmt, aber durchaus prüfen, ob das Datum überhaupt möglich ist. Dabei muß sichergestellt sein, daß Tag und Monat in gültigen Intervallen liegen (Tage: 1 bis 28, 29, 30 oder 31; Monat: 1 bis 12) – außerdem sollte man die Schaltjahre nicht vergessen!

Eine Lösung wäre z.B., die Tageszahl pro Monat in einem Integer-Array zu speichern (etwa DY(1) bis DY(12)) und bei den Schaltjahren den Februar-Wert nach einer IF-THEN-Abfrage um »1« zu erhöhen.

Viel eleganter macht's unser kurzes Basic-Programm:

```
10 INPUT "TAG, MONAT, JAHR";T,M,Y
20 DY=((M-7*INT((M-
1)/7))AND1)+30+(M=2)*(2+
(Y/4=INT(Y/4)))
30 IF T=0 OR M=0 OR T>DY OR M>12 THEN GO-
SUB 100: END
40 PRINT "DIESER MONAT HAT";DY;" TAGE!":
END
100 PRINT "UNZULAESSIGES DATUM!": RETURN
```

Damit ist man sicher, daß der Computer unabsichtlich oder bewußt falsche Datumskonstellationen zurückweist.

## Totaler Black-Out

Wollen Sie den C 64 einmal völlig durcheinander bringen? Dann geben Sie SYS 62391 ein: Ab sofort quittiert er jede per <RETURN> abgeschlossene Eingabe mit der Meldung »Syntax Error« – und wenn sie noch so korrekt ist! Durch <SHIFT CLR/HOME> wird's noch schlimmer: Der Cursor flackert hektisch in der linken oberen Bildschirmecke. Lediglich ein paar Tasten (z.B. <E>) sind noch ansprechbar – ansonsten bleibt der Computer unzugänglich. Will man diese Mißfunktion quasi als Kopierschutz in eigene Programme einbauen, beendet man sie mit dem SYS-Befehl, z.B.:

```
1000 SYS 62319:X
```

Dem unbefugten Anwender ist ein »Overflow Error« sicher. Das Programm im Speicher ist außerdem futsch!

Ohne POKE- oder SYS-Befehle: Mit naturreinem Basic 2.0 bringen Sie den C 64 zum Absturz (getestet auf dem alten C-64-Brotkasten, dem neuen C-64-II und einem C 128 im C-64-Modus):

```
PRINT 5+"A"+-5
```

Auch <RUN/STOP RESTORE> verweigert seine Dienste – da hilft nur noch der Resetknopf oder ausschalten!

## Doppeltes PEEK

Will man 16-Bit-Werte lesen (Low- und High-Byte), kommt man um zwei PEEK-Befehle nicht herum (zwei benachbarte Speicherstellen müssen gelesen werden!). Das High-Byte muß man noch zusätzlich mit »256« multiplizieren.

Beispiel: Die 16-Bit-Adresse fürs Ende eines Basic-Programms befindet sich in den Zeropage-Adressen 45 und 46:

```
PRINT PEEK(45) + 256*PEEK(46)
```

Als Ergebnis erhalten Sie die Nummer der ersten Speicherstelle nach dem letzten Byte des Basic-Programmcodes.

Setzt man aber die DEF FN-Funktion trickreich ein, läßt sich die Basic-Eingabe zur Ermittlung dieser Adresse verkürzen:

```
10 DEF FN DE(X) = PEEK(X) + PEEK(X+1)*256
```

Jetzt reicht der Basic-Befehl:

```
PRINT FN DE(45)
```

Achtung: Die DEF FN-Funktion läßt sich nur innerhalb eines Basic-Programms einrichten, nicht im Direkmodus des C 64!